

上网行为管理 技术分析

——常见 IP 碎片攻击详解

相关主题: 上网行为管理,PatrolFlow,信息安全网关,带宽管理,流量控制,P2P 控制,BT 下载,邮件监控,IM 控制,游戏监管,聊天监控,内容审计,多链路负载均衡,Web 推送,防火墙,防毒墙

正文内容:

常见 IP 碎片攻击详解

本文简单介绍了 IP 分片原理，并结合 Snort 抓包结果详细分析常见 IP 碎片攻击的原理和特征，最后对阻止 IP 碎片攻击给出一些建议。希望对加深理解 IP 协议和一些 DoS 攻击手段有所帮助。

1.为什么存在 IP 碎片

链路层具有最大传输单元 MTU 这个特性，它限制了数据帧的最大长度，不同的网络类型都有一个上限值。以太网的 MTU 是 1500，你可以用 netstat-i 命令查看这个值。如果 IP 层有数据包要传，而且数据包的长度超过了 MTU，那么 IP 层就要对数据包进行分片（fragmentation）操作，使每一片的长度都小于或等于 MTU。我们假设要传输一个 UDP 数据包，以太网的 MTU 为 1500 字节，一般 IP 首部为 20 字节，UDP 首部为 8 字节，数据的净荷（payload）部分预留是 $1500-20-8=1472$ 字节。如果数据部分大于 1472 字节，就会出现分片现象。

IP 首部包含了分片和重组所需的信息：

```

+++++
|Identification|R|DF|MF|FragmentOffset|
+++++
|<-----16----->|<--3-->|<-----13----->|

```

Identification: 发送端发送的 IP 数据包标识字段都是一个唯一值，该值在分片时被复制到每个片中。

R: 保留未用。

DF: Don tFragment, “不分片”位，如果将这一比特置 1，IP 层将不对数据报进行分片。

MF: MoreFragment, “更多的片”，除了最后一块外，其他每个组成数据报的片都要把该比特置 1。

FragmentOffset: 该片偏移原始数据包开始处的位置。偏移的字节数是该值乘以 8。

另外，当数据报被分片后，每个片的总长度值要改为该片的长度值。

每一 IP 分片都各自路由，到达目的主机后在 IP 层重组，请放心，首部中的数据能够正确

完成分片的重组。你不禁要问，既然分片可以被重组，那么所谓的碎片攻击是如何产生的呢？

2.IP 碎片攻击

IP 首部有两个字节表示整个 IP 数据包的长度，所以 IP 数据包最长只能为 0xFFFF，就是 65535 字节。如果有意发送总长度超过 65535 的 IP 碎片，一些老的系统内核在处理的时候就会出现问題，导致崩溃或者拒绝服务。另外，如果分片之间偏移量经过精心构造，一些系统就无法处理，导致死机。所以说，漏洞的起因是出在重组算法上。下面我们逐个分析一些著名的碎片攻击程序，来了解如何人为制造 IP 碎片来攻击系统。

3.pingo death

pingo death 是利用 ICMP 协议的一种碎片攻击。攻击者发送一个长度超过 65535 的 Echo Request 数据包，目标主机在重组分片的时候会造成本先分配的 65535 字节缓冲区溢出，系统通常会崩溃或挂起。ping 不就是发送 ICMP Echo Request 数据包的吗？让我们尝试攻击一下吧！不管 IP 和 ICMP 首部长度了，数据长度反正是多多益善，就 65535 吧，发送一个包：

```
#ping-c1-s65535192.168.0.1
```

```
Error:packet size 65535 is too large. Maximum is 65507
```

不走运，看来 Linux 自带的 ping 不允许我们做坏事。:(

65507 是它计算好的：65535-20-8=65507。Win2K 下的 ping 更抠门，数据只允许 65500 大小。所以你必须找另外的程序来发包，但是目前新版本的操作系统已经搞定这个缺陷了，所以你还是继续往下阅读本文吧。

顺便提一下，记得 99 年有“爱国主义黑客”（“红客”的前辈）发动全国网民在某一时刻开始 ping 某美国站点，试图 ping 死远程服务器。这其实是一种 pingflood 攻击，用大量的 Echo Request 包减慢主机的响应速度和阻塞目标网络，原理和 pingo death 是不一样的，这点要分清楚。

4.jolt2

jolt2.c 是在一个死循环中不停的发送一个 ICMP/UDP 的 IP 碎片，可以使 Windows 系统的机器死锁。我测试了没打 SP 的 Windows2000，CPU 利用率会立即上升到 100%，鼠标无法移动。

我们用 Snort 分别抓取采用 ICMP 和 UDP 协议发送的数据包。

发送的 ICMP 包：

```
01/07-15:33:26.974096192.168.0.9->192.168.0.1
```

```
ICMPTTL:255TOS:0x0ID:1109IpLen:20DgmLen:29
```

```
FragOffset:0x1FFEFragSize:0x9
```

```
080000000000000000.....
```

发送的 UDP 包:

01/10-14:21:00.298282192.168.0.9->192.168.0.1

UDPTTL:255TOS:0x0ID:1109IpLen:20DgmLen:29

FragOffset:0x1FFEFragmentSize:0x9

04D304D20009000061.....a

从上面的结果可以看出:

*分片标志位 MF=0, 说明是最后一个分片。

*偏移量为 0x1FFE, 计算重组后的长度为 $(0x1FFE * 8) + 29 = 65549 > 65535$, 溢出

。

*IP 包的 ID 为 1109, 可以作为 IDS 检测的一个特征。

*ICMP 包:

类型为 8、代码为 0, 是 EchoRequest;

校验和为 0x0000, 程序没有计算校验, 所以确切的说这个 ICMP 包是非法的。

*UDP 包:

目的端口由用户在命令参数中指定;

源端口是目的端口和 1235 进行 OR 的结果;

校验和为 0x0000, 和 ICMP 的一样, 没有计算, 非法的 UDP。

净荷部分只有一个字符 a 。

jolt2.c 应该可以伪造源 IP 地址, 但是源程序中并没有把用户试图伪装的 IP 地址赋值给 `src_addr`, 不知道作者是不是故意的。

jolt2 的影响相当大, 通过不停的发送这个偏移量很大的数据包, 不仅死锁未打补丁的 Windows 系统, 同时也大大增加了网络流量。曾经有人利用 jolt2 模拟网络流量, 测试 IDS 在高负载流量下的攻击检测效率, 就是利用这个特性。

5.teardrop

teardrop 也比较简单, 默认发送两个 UDP 数据包, 就能使某些 Linux 内核崩溃。Snort 抓取的结果如下:

第一个:

01/08-11:42:21.985853192.168.0.9->192.168.0.1

UDPTTL:64TOS:0x0ID:242IpLen:20DgmLen:56MF

FragOffset:0x0FragmentSize:0x24

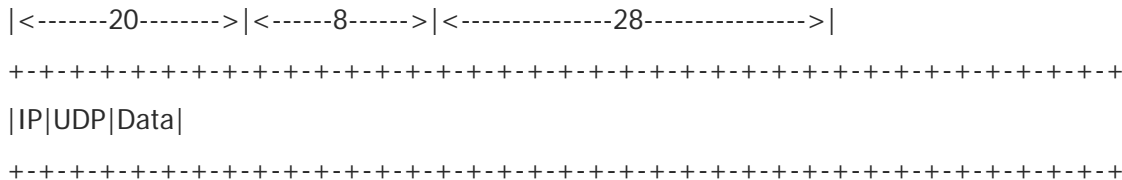
A0A886C700240000000000000000000000.....\$......

00.....

00000000....

*MF=1, 偏移量=0, 分片 IP 包的第一个。

*结构图:



第二个:

01/08-11:42:21.985853192.168.0.9->192.168.0.1

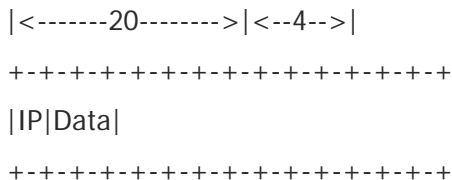
UDPTTL:64TOS:0x0ID:242IpLen:20DgmLen:24

FragOffset:0x3FragSize:0x4

A0A886C7....

*MF=0, 偏移量=0x3, 偏移字节数为 $0x3*8=24$, 最后一个分片。

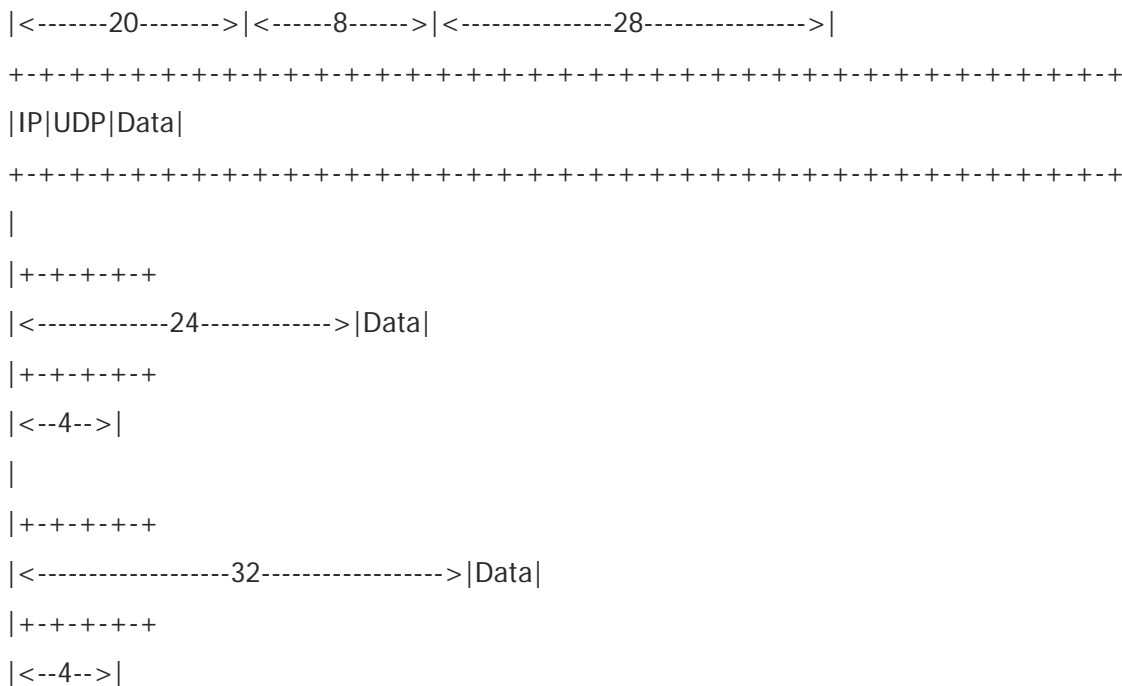
*结构图:



如果修改源代码, 第二片 IP 包的偏移量也可以为 0x4, 偏移字节数就是 $0x4*8=32$ 。

下面的结构图表示了接收端重组分片的过程, 分别对应于偏移字节数为 24 和 32 两种情况

:



可以看出，第二片 IP 包的偏移量小于第一片结束的位移，而且算上第二片 IP 包的 Data，也未超过第一片的尾部，这就是重叠现象（overlap）。老的 Linux 内核（1.x-2.0.x）在处理这种重叠分片的时候存在问题，WinNT/95 在接收到 10 至 50 个 teardrop 分片时也会崩溃。你可以阅读 teardrop.c 的源代码来了解如何构造并发送这种数据包。

6.如何阻止 IP 碎片攻击

*Windows 系统请打上最新的 ServicePack，目前的 Linux 内核已经不受影响。

*如果可能，在网络边界上禁止碎片包通过，或者用 iptables 限制每秒通过碎片包的数目。

*如果防火墙有重组碎片的功能，请确保自身的算法没有问题，否则被 DoS 就会影响整个网络。

*Win2K 系统中，自定义 IP 安全策略，设置“碎片检查”。

更多信息请登录 <http://www.byzoro.com>